Министерство науки и высшего образования

Российский Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

ДВОИЧНАЯ АРИФМЕТИКА

Отчёт по лабораторной работе №1

по дисциплине «Вычислительные системы, сети и телекоммуникации»

|  |  |
| --- | --- |
|  | Обучающийся гр. \_\_\_\_\_\_\_442-1\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кадочников А.В.  (подпись) (И.О. Фамилия)  \_\_\_\_22.09.23\_\_\_  (дата) |
| **\_\_\_\_\_\_\_\_\_**  (оценка) | Принял  Доцент АСУ  (должность, ученая степень, звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Алфёров С.М.  (подпись) (И.О. Фамилия)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) |

Томск 2023

Оглавление

[**1 Цель работы** 3](#_Toc146288613)

[**2 Задание** 4](#_Toc146288614)

[**3 Текст программы** 5](#_Toc146288615)

[**4 Результаты работы программы при разных входных данных** 6](#_Toc146288616)

[**5 Выводы** 8](#_Toc146288617)

# **1 Цель работы**

Получить представление о способах хранения числовых данных в памяти ЭВМ.

# **2 Задание**

Написать программу по выводу битовых данных (битов), числовых значений, хранящихся в переменных. Задать значения и ввести на экран биты: целых знаковых и беззнаковых чисел длиной 1,2 и 4 байта; вещественных чисел с плавающей запятой длиной 4 и 8 байт. Объяснить результат.

# **3 Текст программы**

#include <stdio.h>

void print\_bit(void \*ptr, size\_t size) {

    unsigned char \*p = (unsigned char\*)ptr;

    for (int i = size - 1; i >= 0; i--) {

        for (int j = 7; j >= 0; j--) {

            printf("%d", (p[i] >> j) & 1);

        }

        printf(" ");

    }

    printf("\n");

}

int main() {

    char c = -1;

    short s = -1;

    int i = 1024;

    unsigned char uc = 1;

    unsigned short us = 1;

    unsigned int ui = 1;

    float f = 0.15625;

    double d = 0.15625;

    printf("Целые знаковые числа:\n");

    printf("char: %d\n", c); //1

    print\_bit(&c, sizeof(c));

    printf("short: %d\n", s); //2

    print\_bit(&s, sizeof(s));

    printf("int: %d\n", i); //4

    print\_bit(&i, sizeof(i));

    printf("\nЦелые беззнаковые числа:\n");

    printf("unsigned char: %d\n", uc);//1

    print\_bit(&uc, sizeof(uc));

    printf("short: %d\n", us); //2

    print\_bit(&s, sizeof(us));

    printf("unsigned int: %d\n", ui);

    print\_bit(&ui, sizeof(ui)); //8

    printf("\nДробные числа с плавающей точкой:\n");

    printf("float: %f\n", f); //4

    print\_bit(&f, sizeof(f));

    printf("double: %f\n", d); //8

    print\_bit(&d, sizeof(d));

    return 0;

}

# **4 Р езультаты работы программы при разных входных данных**

Проверка правильности работы:

Float: 1 бит под знак, 8 битов под порядок – порядок, 23 бита на мантиссу  
-1.0 – 00111111100000000000000000000000 ->   
Вначале 1, значит число отрицательное, порядок – 01111111 = 127  
Т.к порядок записан в нормализованном виде нужно вычесть 127, 127-127 = 0  
2^0 = 1.0 ответ сошелся.

Double: 1 бит знак, 11 бит порядок, мантисса – 52 бита   
10.2– 0100000000100100011001100110011001100110011001100110011001100110 ->  
Вначале 0, потому что число положительное  
Порядок:

23 <= 10, 1023+3 = 1026 -> 10000000010 – порядок.

10 = 10102

0.2\*2=0.4(0)  
0.4\*2=0.8(0)  
0.8\*2=1.6(1)  
0.6\*2=1.2(1)  
…  
1010.0011001100110011001100110011001100110011001100110  
Сдвинем число на 3 разрядов вправо получим мантиссу  
М = 1.0100011001100110011001100110011001100110011001100110  
так как первую единицу мантиссы не записывают, то по итогу получаем:  
0100000000100100011001100110011001100110011001100110011001100110  
Ответ сошёлся.

**5 Выводы**

В процессе работы над данной лабораторной работой я разобрался в способах хранения числовых данных в памяти ЭВМ и научился их представлять с помощью программы.